

# CalcSheet

## Instrukcja użytkownika

### Obsługa interfejsu graficznego

Interfejs graficzny programu CalcSheet składa się z paska menu, pola tekstowego, przycisku wprowadzania zmian oraz zestawu dostępnych komórek. Umożliwia on podane czynności:

- zastąpienie obecnego arkusza nowym,
- wczytanie arkusza z pliku,
- zapisanie arkusza do pliku,
- edycję zawartości komórki
- zamknięcie programu.

#### ZEROWANIE, WCZYTYWANIE I ZAPISYWANIE ARKUSZA

Wykonywanie operacji na arkuszu umożliwiają poszczególne zakładki w karcie *File* znajdującej się na górnym pasku menu. Przy wczytywaniu i zapisywaniu arkuszy program korzysta z formatu `*.ser`.

#### EDYCJA ZAWARTOŚCI KOMÓREK

Poprzez pole tekstowe: aby edytować zawartości komórki, należy ją zaznaczyć, a następnie edytować jej zawartości za pomocą pola tekstowe. Zmiany należy zatwierdzić przyciskiem znajdującym się obok pola tekstowego.

Za pomocą funkcji *drag&drop*: by skopiować wartość komórki do określonego prostokąta należy przeciągnąć jego lewy górny róg na prawy dolny. Spowoduje to umieszczenie wartości z lewego górnego wierzchołka w każdej komórce znajdującej się w prostokącie. *Uwaga*: Kopiowanie powoduje odpowiednie zwiększanie liczb / adresów komórek występujących w wyrażeniu. Zachowanie to jest domyślne, ale opcjonalne (szczegóły poniżej).

### Dostępne wyrażenia

Program wyposażony jest w interpreter prostego języka programowania, dzięki czemu umożliwia wyliczanie złożonych wyrażeń przy zachowaniu naturalnego priorytetu i łączności operatorów. Poniżej znajduje się lista wyrażeń poprawnie interpretowanych przez program.

#### LICZBY, WARTOŚCI BOOLOWSKIE, STAŁE I ZMIENNE

Poprawnymi wyrażeniami są `m.in.` liczby zmiennoprzecinkowe, stałe logiczne, tzn. wartości `true` , `false` , oraz dostępne stałe liczbowe (obecnie  $e$  oraz  $\pi$ ). Ponadto wyrażeniami są odwołania do innych komórek w arkuszu. Komórki adresowane są przez literę odpowiadającą kolumnie oraz liczbę odpowiadającą wierszowi (np. pierwsza komórka pierwszego wiersza ma adres "A1").

## OPERATORY

W języku programu dostępne są operatory różnych priorytetów, łączności, arności oraz typów. Poniższa tabela zawiera szczegóły.

Operator	Arność	Priorytet	Łączność	Typ wejściowy	Typ wyjściowy
<, >, <>, <=, >=, =	2	1	brak	liczba	bool
+, -	2	2	w lewo	liczba	liczba
	2	2	w lewo	bool	bool
*, /, %	2	3	w lewo	liczba	liczba
&	2	3	w lewo	bool	bool
^	2	4	w prawo	liczba	liczba
-	1	5	łączny	int	int
~	1	5	łączny	bool	bool

## WYRAŻENIA WARUNKOWE

Wyrażenia postaci `if e1 then e2 else e3` gdzie `e1`, `e2`, `e3` są wyrażeniami.

## WYWOŁANIA FUNKCJI

Program zawiera definicję kilku standardowych funkcji (obecnie *sin*, *cos* oraz *exp*), które wywołać można w następujący sposób: `fun x` lub `fun(y)`. Wywołanie funkcji wiąże silniej niż wszystkie operatory, zatem np. instrukcja `exp 2 + 2` odpowiada wyrażeniu  $e^2 + 2$ . Dodatkowo, wywołania funkcji łączą w prawo, więc np. `sin cos x` odpowiada wyrażeniu  $\sin(\cos(x))$ .

## OPERACJE NA PROSTOKĄTACH

Program umożliwia również wykonywanie różnych operacji na prostokątach - np. sumowania wartości, czy zliczania komórek, które spełniają dany warunek. Odbywa się to za pomocą instrukcji `apply (range, lambda)`, gdzie `range` to zakres prostokąta, na którym ma być wykonana operacji (podany w formie `x:y`, gdzie `x`, `y` to nazwy odpowiednio lewego górnego oraz prawego dolnego wierzchołka), a `lambda` jest wyrażeniem, które należy obliczyć na każdej komórce z zakresu. Wyniki poszczególnych obliczeń są następnie sumowane. Lambdy zapisuje się w formie `x -> e`, gdzie `x` jest dowolną nazwą zmiennej, pod którą podstawiona zostanie wartość komórki, a `e` jest wyrażeniem, które chcemy obliczyć (prawdopodobnie zależnym od zmiennej `x`).

## BLOKOWANIE INKREMENTACJI WARTOŚCI / ADRESÓW

Domyślnym zachowaniem programu przy przeciąganiu wyrażeń jest inkrementacja znajdujących się w nich liczb / adresów zmiennych. Aby wyłączyć to zachowanie dla konkretnej liczby / zmiennej należy dopisać przed jej wystąpieniem operator `$`.

## Wyliczanie wartości komórek

Po zakończeniu edycji wyrażenia w komórce następuje wyliczenie jej wartości. Pierwszym etapem tego procesu jest kontrola typów, podczas której sprawdzane jest *m.in.* czy wszystkie komórki / funkcje, do których odwołuje się wyrażenie istnieją i są zdefiniowane, czy funkcjom i operatorom podane są odpowiednie typy, itd. Jeżeli program nie przejdzie kontroli typów, zostanie wyświetlony błąd, a obliczenie wartości komórki zostanie odroczone.

Po zakończonej kontroli typów następuje faktyczne obliczenie wartości komórki. Gdy znana jest jej dokładna wartość, następuje ponowne obliczenie komórek, które są od niej zależne (a następnie tych, które zależne są od nich, itd.).

*Uwaga:* nie dopuszczamy występowania cyklicznych zależności między komórkami. W przypadku wykrycia takiej sytuacji, zostanie wyświetlony błąd, a obliczenie zostanie odroczone.

Po obliczeniu wartości wszystkich zależnych komórek ich zawartość jest ponownie wyświetlana na ekran.